2-Day Course – Spatial Modeling with Geostatistics

Prof. Michael J. Pyrcz, Ph.D., P.Eng.
Associate Professor

Hildebrand Department of Petroleum & Geosystems Engineering
University of Texas at Austin

Bureau of Economic Geology, Jackson School of Geosciences
University of Texas at Austin

“In two days, what a geoscientists needs to know about geostatistics, and workflows to get you started with applying geostatistics to impact your work.”

Spatial Modeling with Geostatistics

Machine Learning

Lecture outline . . .

- **Inference and Prediction**
- **Prediction Accuracy**
- **Decision Tree**

Additional Resources

James, G, Witten, D., Hastie, T. and Tibshirani, R., 2013, An Introduction to Statistical Learning with Applications in R, Springer, New York

Prerequisites

Introduction

Probability Theory

Representative Sampling

Spatial Data Analysis

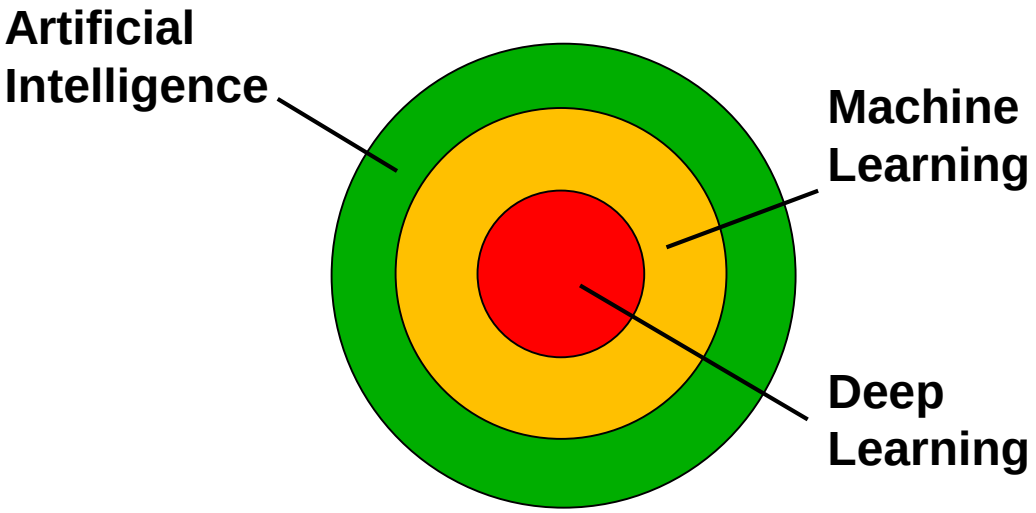
Spatial Estimation

Stochastic Simulation

Uncertainty Management

Machine Learning

Machine Learning / Statistical Learning



Artificial Intelligence: the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages (Google Dictionary)

Machine Learning: is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed (Google Dictionary). Access data and learn for themselves.

Deep Learning: subset of machine learning for unsupervised learning from unstructured, unlabeled data.

Machine Learning / Statistical Learning

Big Data: you have big data if your data has a combination of these:

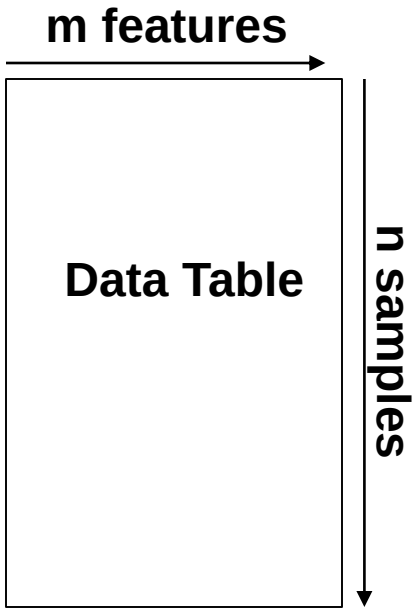
Volume: large number of data samples, large memory requirements and difficult to visualize

Velocity: data is gathered at a high rate, continuously relative to decision making cycles

Variety: data form various sources, with various types and scales

Variability: data acquisition changes during the project

Veracity: data has various levels of accuracy



“Energy has been big data before tech learned about big data.”

Big Data Analytics – methods to explore and detect patterns, trends and other useful information from big data to improve decision making.

Machine Learning / Statistical Learning

To better utilize data to improve decision making with consistency and speed.

- Applications in Energy
  1. Feature detection / Guided interpretation in dense data sets like seismic, smart fields / Big data analytics
  2. Optimization of field development decisions
  3. Exploration prioritization

- Why is Energy different?
  - sparse and uncertain data
  - complicated and heterogeneous systems
  - high degree of irreducible interpretation

Machine Learning / Statistical Learning

- Just like spatial statistics / geostatistics, statistical learning is a set of tools to add to your tool box as an engineer

- Each is very dangerous to use as a black box. You will need to understand what's under the hood
  - methods, workflows, assumptions and limitations.
  - scope and trade offs between alternative methods

- Imagine you are a carpenter (all geostatistics workflows) (Pyrcz and Deutsch, 2014).
  - You would have a tool box
  - You would know each tool perfectly well
  - Understand performance over a variety of applications
  - You would understand the range of applications, weaknesses, strengths, limits.
  - Choice between tools would be based on expert judgement of circumstances and goals of a project
  - You would choose specific tools to have ready for use and other for more rare circumstances
  - Too few tools and a box overwhelmed with obscure tools are both issues.

The Model

- Predictors, Independent Variables, Features
  - input variables
  - for a model $Y = f(X_1, \ldots, X_m) + \epsilon$, these are the $X_1, \ldots, X_n$
  - note ϵ is a random error term
- Response, Dependent Variables
  - output variable
  - for a model $Y = f(X_1, \ldots, X_m)$, this is Y
- Statistical Learning is All About
  - Estimating f for two purposes

1. Prediction
   - $\hat{Y} = \hat{f}(X)$
   - where $\hat{f}$ is the estimate of f and $\hat{Y}$ is the resulting prediction of Y
2. Inference
   - $\widehat{\partial Y} = \hat{f}(\partial X)$
   - relationship (exists?, form?) between each predictor and the response

Prediction

- Accuracy of $\hat{Y}$ depends on reducible and irreducible error
  - $\hat{f}$ is not a perfect model. Error due to the estimate of f is reducible error
  - but even if we had f, $\hat{Y} = f(X)$, prediction would still have error
  - This is because Y is a function of ϵ, $Y = f(X) + \epsilon$
  - and ϵ is irreducible

- What is irreducible error?
  - includes unmeasured variables that would be useful for predicting Y
  - includes unmeasured variation $X_\alpha < x_{\alpha_{min}}$ or $X_\alpha > x_{\alpha_{max}}$

- Estimation Error
  - We can use the concept of estimation error

$$E\left[\left(Y - \hat{Y}\right)^2\right] = E\left[\left(f(X) + \epsilon - \hat{f}(X)\right)^2\right]$$

$$\underbrace{E\left[\left(f(X) - \hat{f}(X)\right)^2\right]}_{\text{Reducible}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}}$$

Inference

- There is value in understanding the relationships
  - for $Y = f(X_1, \ldots, X_m) + \epsilon$ we can understand the influence of each X_α on Y

1. Which predictors are associated with the response?
   a) What data to collect? Value of information.
   b) What data to focus on? Simplification of the model. Communication. Big hitters.

2. What is the relationship between each response and each predictor?
   a) sense of the relationship (positive or negative)?
   b) shape of relationship (sweet spot)?
   c) relationships may depend on values of other predictors!

3. Can the relationship be modeled linearly?
   a) much simplified
   b) very low parametric representation
   c) use multiGaussian?

Estimating f

- Parametric Methods
  - make an assumption about the functional form, shape
  - use training data to fit or train the model
  - test the model with withheld test data
  - for example, here is a linear model
  - there is a risk that $\hat{f}$ is quite different than f, then we get a poor model!

$$Y = f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m$$

- Linear Model
  - an equation of the first degree in any number of variables
  - requires only quadratic minimization to solve for the coefficients

- Model fitting
  - Apply training data
  - Solve for (ordinary) least squares solution for coefficients β_0, β_1,…, β_m

Estimating f

- Nonparametric Methods
  - make no assumption about the functional form, shape
  - estimate f that approaches the data without being too rough
  - more flexibility to fit a variety of shapes for f
  - less risk that $\hat{f}$ is a poor fit for f
  - do not reduce the problem to estimating a small set of parameters; therefore, typically need a lot more data for an accurate estimate of f

Prediction Accuracy vs. Model Interpretability

- Prediction Accuracy
  - may be measured as the estimation error

$$E\left[\left(Y-\hat{Y}\right)^2\right]$$

  - improves with the complexity of the model
  - flexibility to fit the data
  - for example, a linear model may not fit the available data as well as an artificial neural net!

- Interpretability
  - is the ability to understand the model
  - how each predictor is associated with the response
  - for example, with a linear model is very easy to observe the influence of each predictor on the response
  - but for an artificial neural net it is very difficult

Regression vs. Classification

- Regression
  - the response variable is continuous
- Classification
  - the response variable is categorical (in machine learning terms known as qualitative)

Assessing Model Accuracy

- Method Selection is Important
  - No one method performs well on all datasets.
  - Based on experience, understanding the data and limitations of the methods

- Measuring Quality of Fit
  - for regression, the most common measure is the mean square error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left[\left(y_i - \hat{f}(x_1^i, \ldots, x_m^i)\right)^2\right]$$

where we have n observations. The challenge is that that real question we have is how well can we predict at an unsampled location.

$$\mathrm{E}\left[\left(y_0 - \hat{f}(x_1^0, \ldots, x_m^0)\right)^2\right]$$

over a variety of unsampled sets of predictors $x_1^0, \ldots, x_p^0$. We want to know how our model performs when we move away from the training set of data!

Bias and Variance Trade-off

Michael Pyrcz, the University of Texas at Austin, @GeostatsGuy

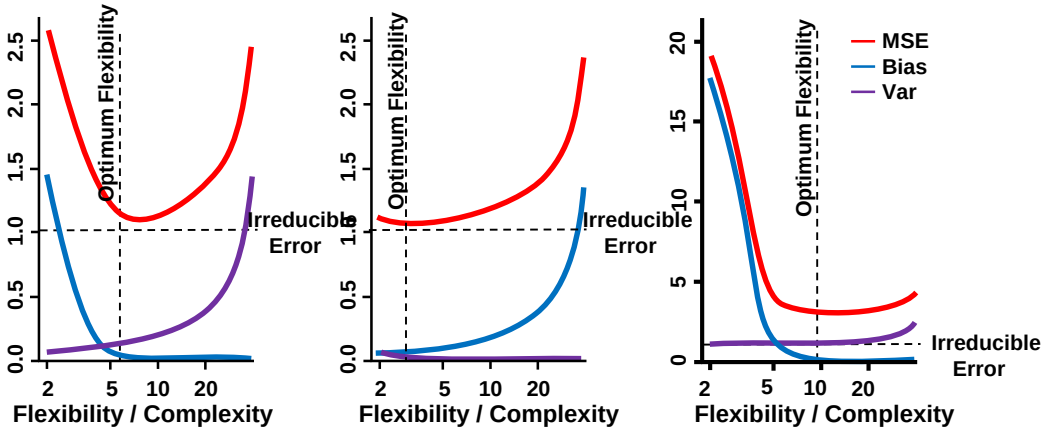
- The **Expected Test Mean Square Error** may be calculated as:

$$\mathrm{E}\left[\left(y_0 - \hat{f}(x_1^0, \ldots, x_m^0)\right)^2\right] = \underbrace{Var\left(\hat{f}(x_1^0, \ldots, x_m^0)\right)}_{\text{Model Variance}} + \underbrace{\left[Bias\left(\hat{f}(x_1^0, \ldots, x_m^0)\right)\right]^2}_{\text{Model Bias}} + \underbrace{Var(\epsilon)}_{\text{Irreducible}}$$

Model Variance is the variance if we had estimated the model with a different training set (simpler models ⇩ lower variance)

Model Bias is error due to using an approximate model (simpler models ⇧ higher bias)

Irreducible error is due to missing variables and limited samples ⇨ can't be fixed with modeling



Examples of model variance, model bias and test MSE for 3 datasets with variable flexibility.

Modified from James, G, Witten, D., Hastie, T. and Tibshirani, R., 2013, An Introduction to Statistical Learning with Applications in R, Springer, New York

Now We Begin Machine Learning

- With these concepts established, let's start to get into machine learning / statistical learning methods
  - These methods will allow you to perform inference and prediction
  - Work with complicated data sets / big data analytics
  - Detect patterns in data

- Remember in our business to win:
  - Have the best data
  - Use the data best

- We are at the beginning of the 4th paradigm for scientific discovery
  - Data-drive discovery

- Smart fields, 4D seismic surveys, computational resources
  - Expanding opportunities for machine learning

- We'll start unsupervised, dimensional reduction:
  - Principal Component Analysis

Decision Trees

- Decision trees are used for supervised learning.

$$Y = f(X_1, \ldots, X_m) + \epsilon$$

  we are predicting a response, Y, from a set of features, $X_1, \ldots, X_m$
- May work with continuous Y for regression or categorical Y for classification.

- Why cover decision trees?
  - They are not the most powerful, cutting edge method in machine learning
  - But they are likely the most understandable, interpretable
  - Decision trees are expanded with random forests, bagging and boosting to be cutting edge.

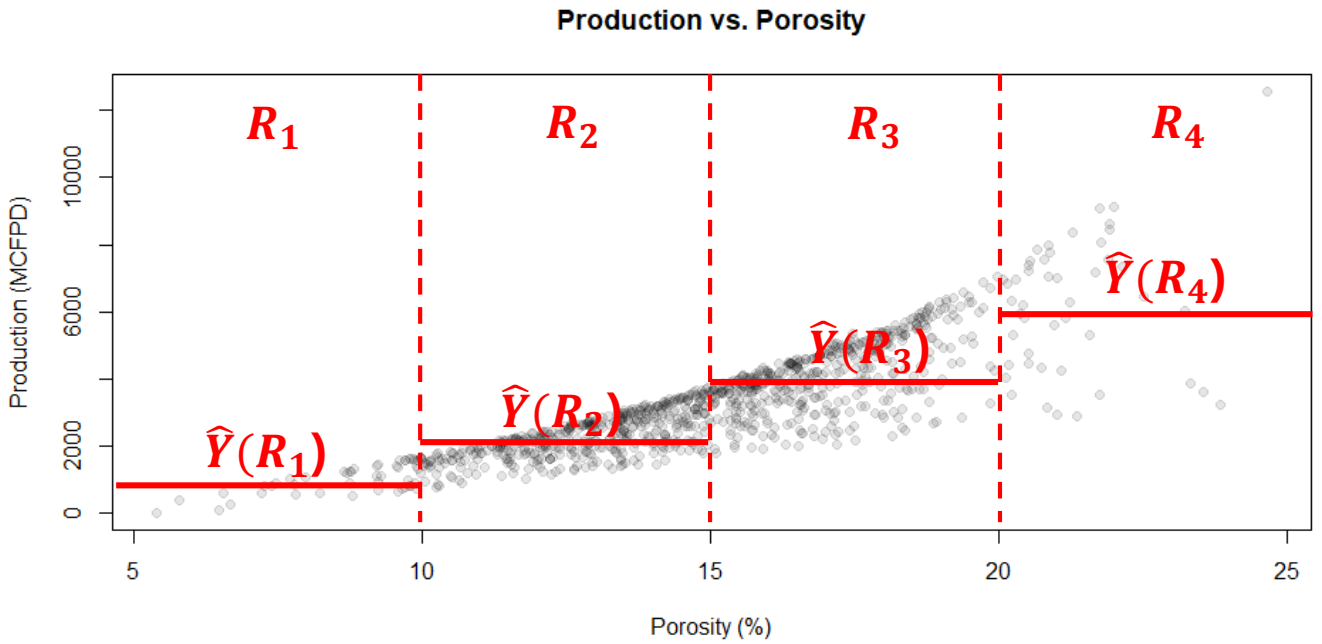
  - “Let’s learn first about a single tree and then we can comprehend the forest.”

Decision Trees

- The fundamental idea is to divide the predictor space, $X_1, \ldots, X_m$, into J mutually exclusive, exhaustive regions

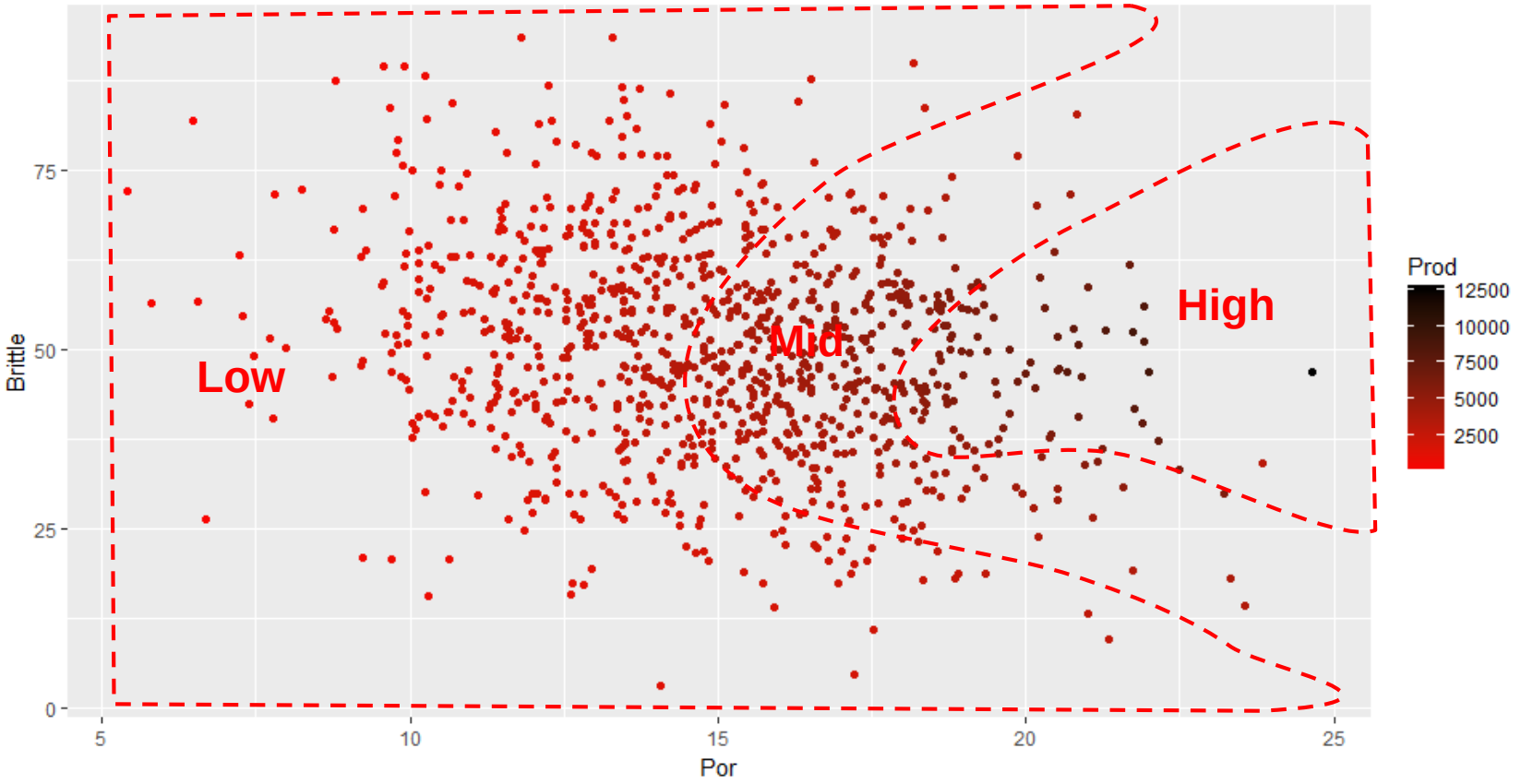
  - mutually exclusive – any combination of predictors only belongs to a single region, R_j
  - exhaustive – all combinations of predictors belong a region, R_j

- For every observation in a region, R_j, we use the same prediction



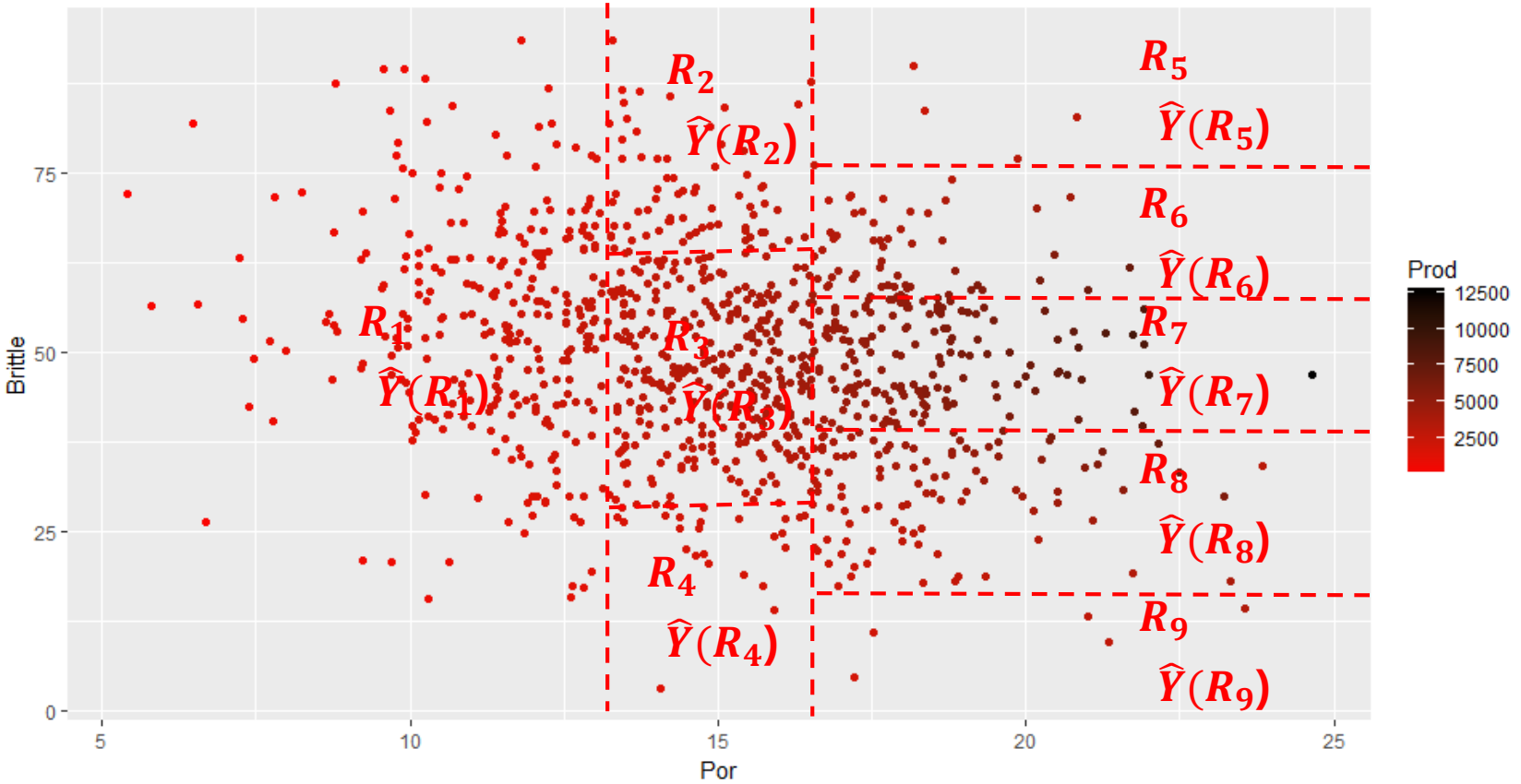
Decision Trees – The Regions

- How do we construct the Regions, $R_1, R_2, \ldots, R_J$?
  - They could be any shape!
  - Consider the 3 variable problem below.
- Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)



Decision Trees – The Regions

- How do we construct the Regions, $R_1, R_2, \ldots, R_J$?
  - They could be any shape!
  - Consider the 3 variable problem below.
  - We decide to use high-dimensional rectangles or boxes ⇨ simple interpretation / rules
    - » Hierarchical segmentation over the features.



Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)

Decision Trees – The Regions

How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

- We want to minimize the Residual Sum of Squares:

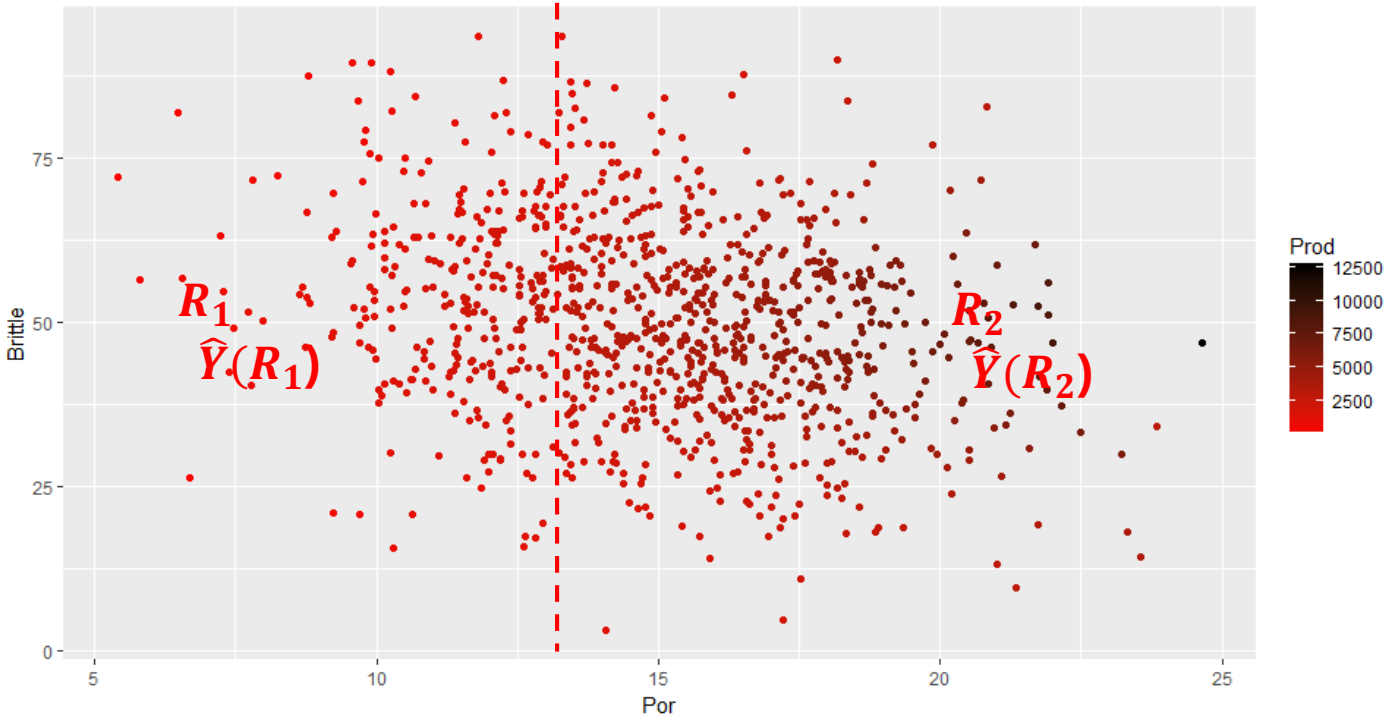
$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

- This is the sum of squares of all the data vs. the estimate in their region (the mean of the training data in the region)
- Hint: somehow we need to account for the cost of complexity
  - » We do this through cross validation and pruning

Decision Trees – The Regions

How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

- Recursive, binary splitting
  - Greedy
    - » at each step the method selects the choice that minimizes RSS. There is no attempt to look ahead, jointly optimize over multiple choices
  - Top-down
    - » at the beginning all data belong to a single region, top of the tree
    - » greedy selection of the single best split over any feature that best reduces the RSS



Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)

Decision Trees – The Regions

How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

- Let's start with one region with all the training data in it
  - We will place the region boundaries based on a threshold, s, inside a previous region, j, such that they minimize the RSS.
  - This requires search over all possible thresholds over all features
  - This is computationally not impossible

$$R_{1(m,s)} = \{X | X_m < s\} \text{ and } R_{2(m,s)} = \{X | X_m \geq s\}$$

  - X_m are the features and s is the threshold for the segmentation into R_1 and R_2

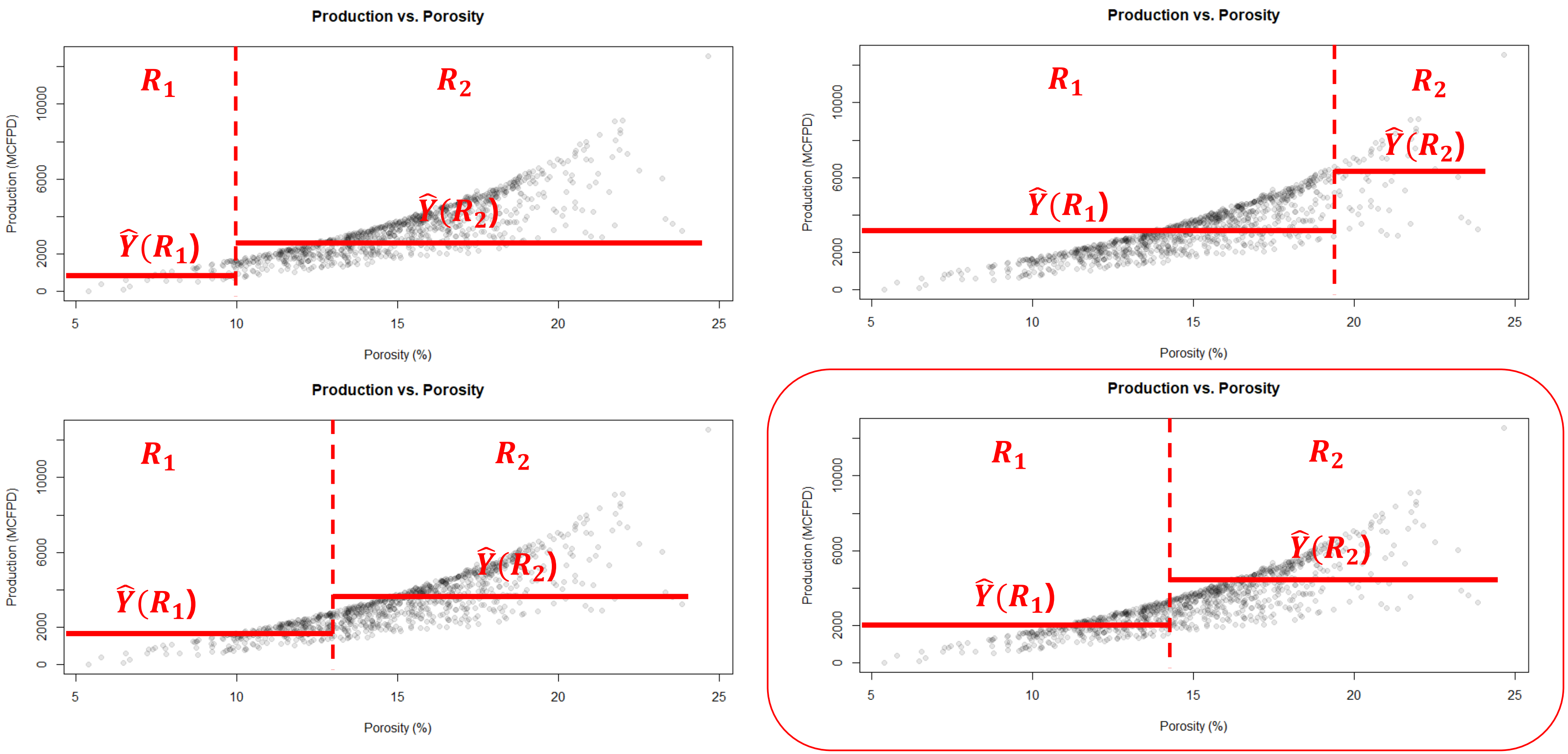
  - We segment such that we minimize the Residual Sum of Squares:

$$RSS = \sum_{i:x_i \in R_1(m,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(m,s)} (y_i - \hat{y}_{R_2})^2$$

Decision Trees – The Regions

Let's pause and go back to our initial bivariate problem and make a tree by hand!

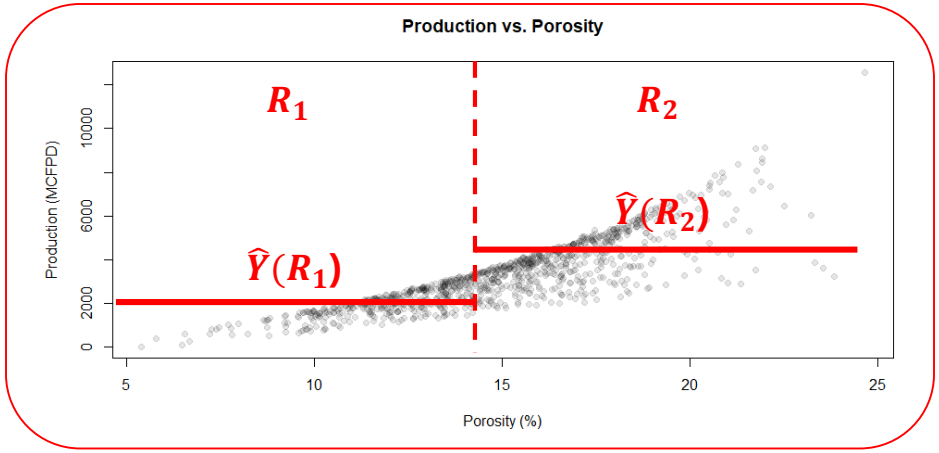
- Where should we split to minimize the error in a tree-based estimate (minimize the residual sum of square)?

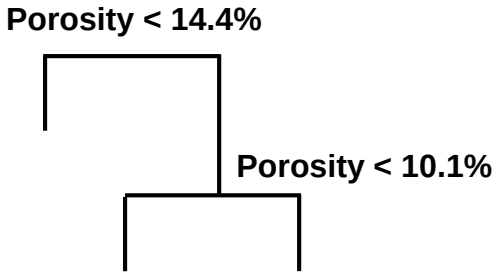


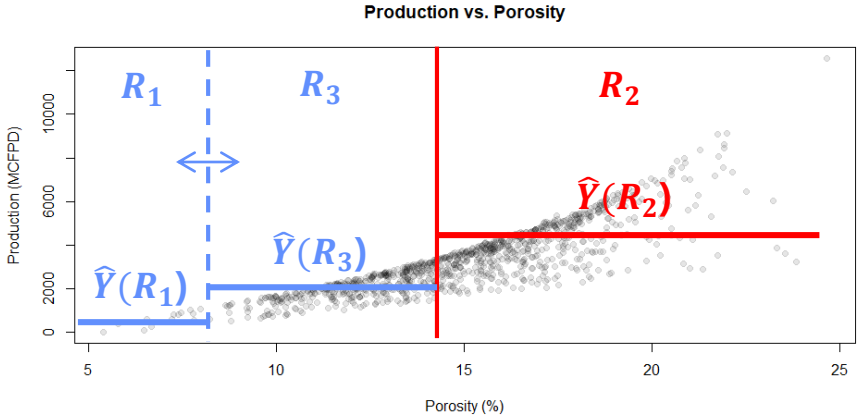
Decision Trees – The Regions

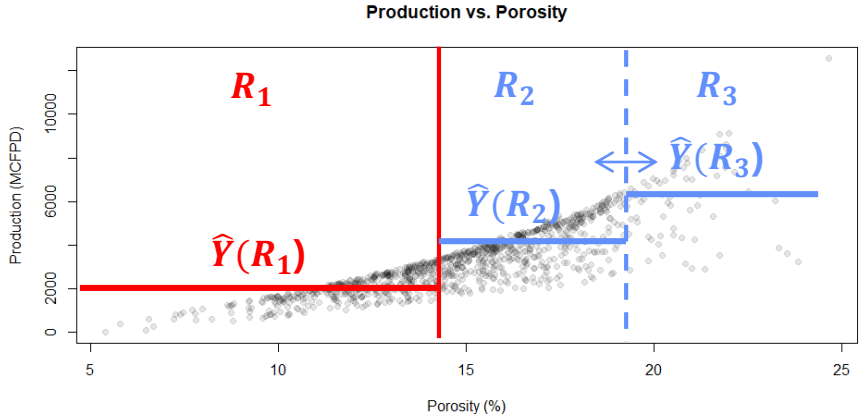
Let's pause and go back to our initial bivariate problem and make a tree by hand!

- Found first split, now check for next split the maximizes accuracy







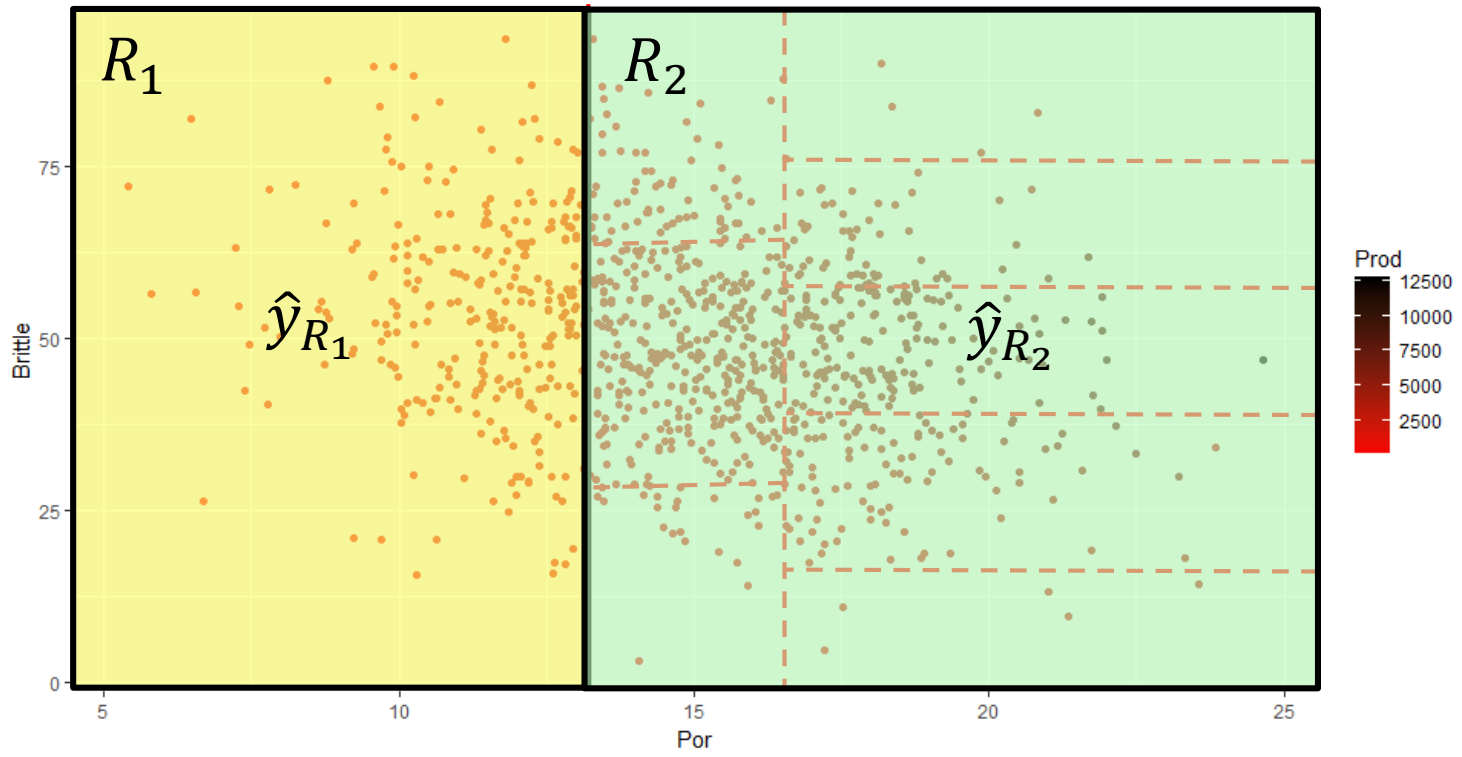


Decision Trees – The Regions

How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

- The we continue sequentially segmenting region with threshold.
  - We will place the region boundaries based on a threshold, s, inside a previous

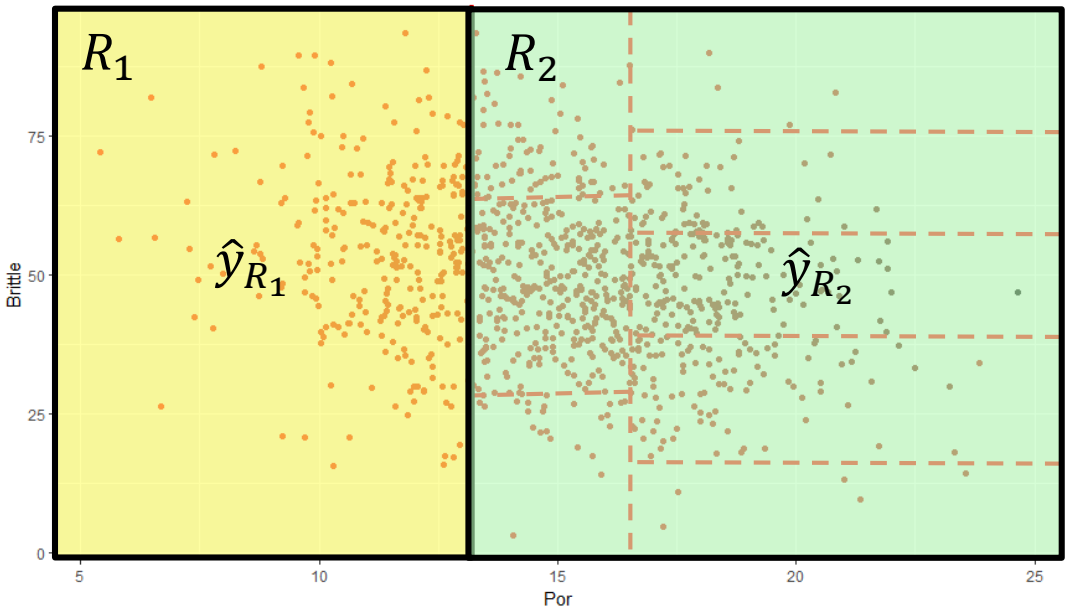
$$RSS = \sum_{i:x_i \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2} (y_i - \hat{y}_{R_2})^2 + \cdots + \sum_{i:x_i \in R_J} (y_i - \hat{y}_{R_J})^2$$

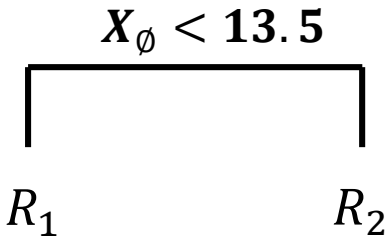


Prediction of unconventional well production (MCFPD) from porosity (%) and brittleness (%)

Decision Trees – The Regions

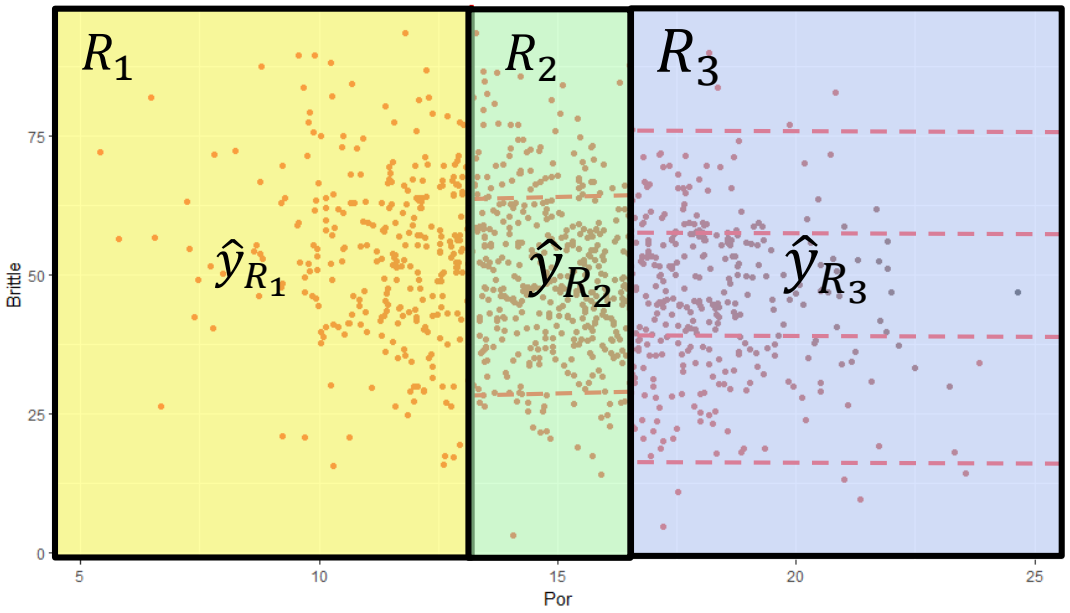
How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

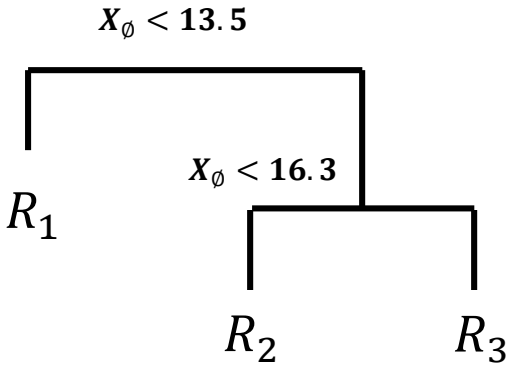




Decision Trees – The Regions

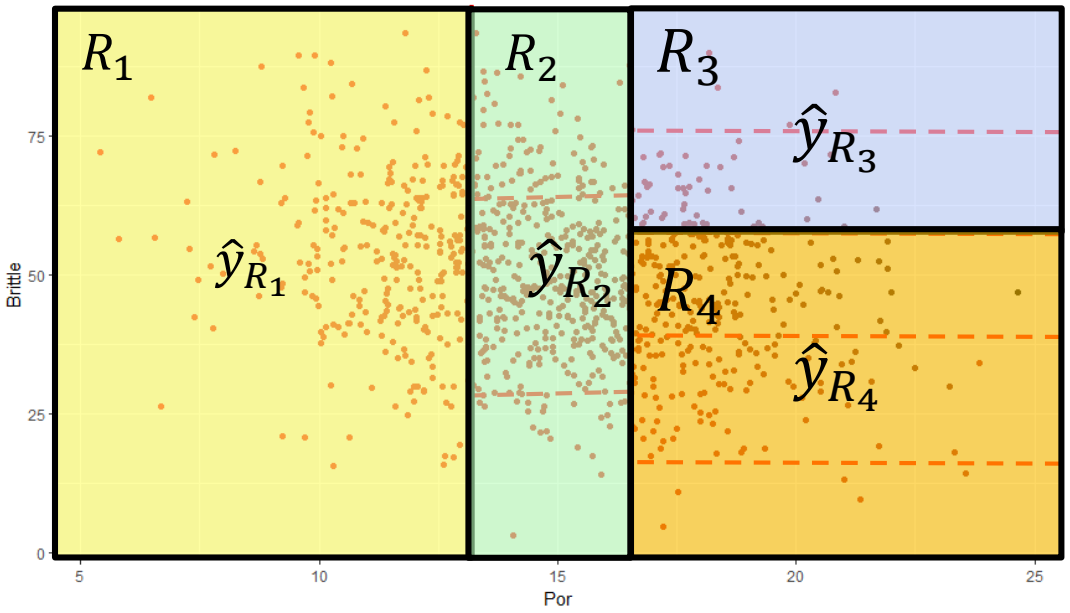
How do we construct the Regions, $R_1, R_2, \ldots, R_J$?

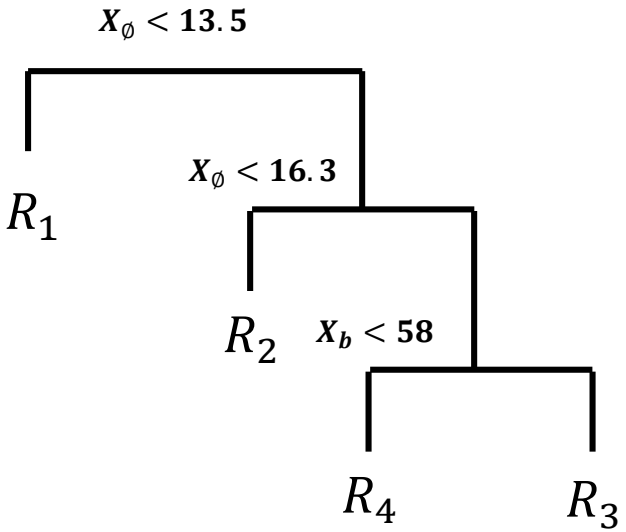




Decision Trees – The Regions

How do we construct the Regions, $R_1, R_2, \ldots, R_J$?



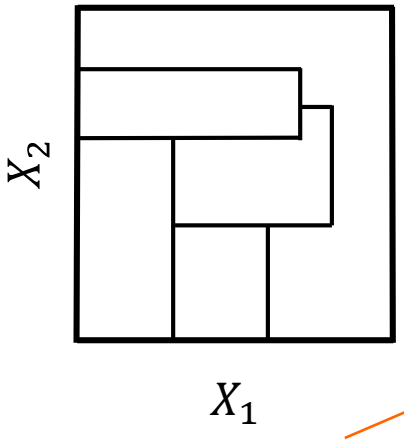


Decision Trees – The Regions

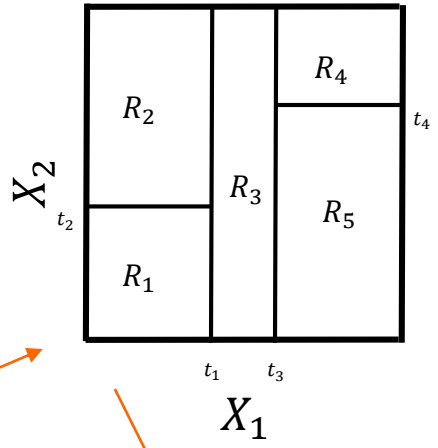
Examples of methods to segment the solution space.

- Top-left 2D feature space partitioning that could not result from recursive binary splitting
- Top-right feature space partitioning, decision tree and estimation surface for feature space.

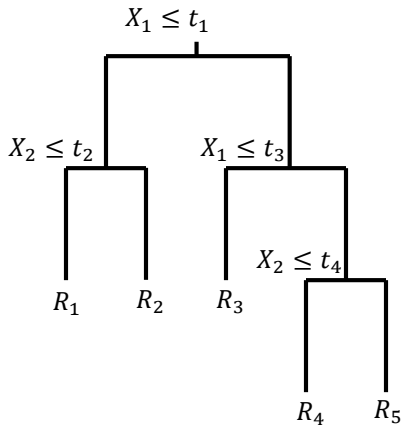
Not from recursive binary splitting



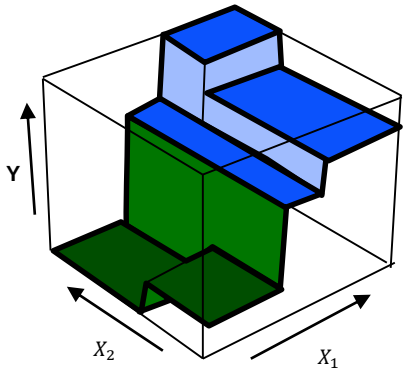
Segmented Feature Space



Decision Tree



Prediction Surface



Decision Trees – Termination

When do we stop recursive binary splitting?

- We could continue until every training data value is in it's own box!
  - This would be over fit!

- The typical approach is to apply a minimum training data in each box criteria
  - The algorithm stops when all boxes have reached the minimum

- We could continue until we cannot not significantly reduce RRS
  - But the current split could lead to an even better split ⇨ short sighted

Decision Trees – Pruning

Why do we want a less complicated tree?

- Decision trees, if allowed to grow complicated are generally overfit.
- It is better to simplify the tree to a smaller tree with fewer splits
  - » lower model variance
  - » better interpretation
  - » with little added model bias
- Limiting tree growth with a high decrease in RSS hurtle is short sighted
- Best strategy is to build a large, complicated tree and then to prune the tree.
  - » We then select the sub tree to provides the lowest test error rate
  - » We cannot consider all possible sub trees (too vast of a solution space)

Decision Trees – Steps

Building a Regression Tree

1. Apply recursive binary splitting to grow a large tree with training data, stop when each terminal node has fewer than a minimum number of data or insufficient RSS decrease.
2. Obtain the sequence of best subtrees as a function of complexity (number of terminal nodes) and RSS with training.
3. Use k-fold cross validation to chose the best complexity value. Divide the training observations into K folds. For each fold, $k = 1, \ldots K$:
   a) Repeats steps from 1-2 on all training excluding those in k fold.
   b) Evaluate the RSS on the left out data in the k fold.
4. Average the error for each α (K results over each fold) and select complexity (number of terminal nodes) that provides low enough RSS.

K-fold Cross Validation

Cross Validation

- Withhold subset of the data during model training
- Then testing the trained model with withheld subset dataset
- Must make sure cross validation is fair
- Training data set (used for training), Testing data set (withheld for testing)

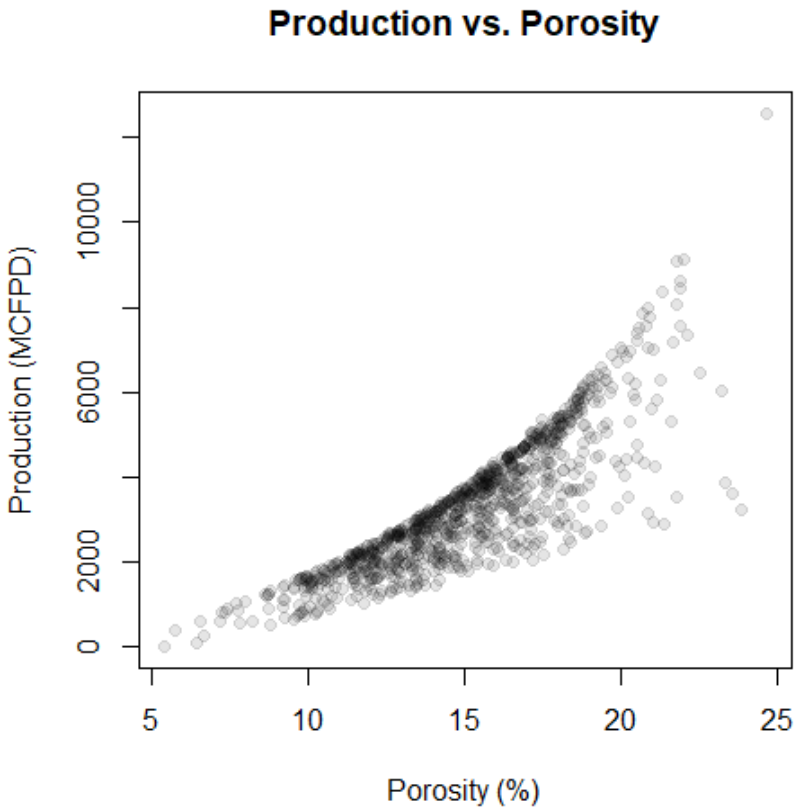
K-fold Approach

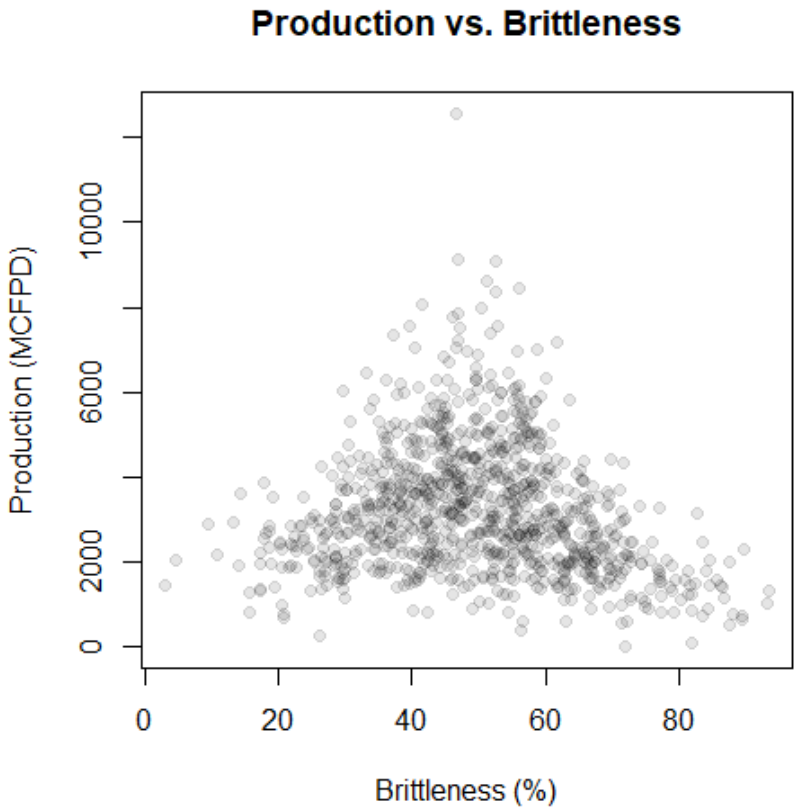
- Select K, for example
- Break data set into K subsets
- Loop over K subsets:
  - use data outside the K part to predict inside the K subset
- Average to summarize the result

Decision Tree – Demonstration in R

Let's use our Unconventional Multivariate Data

- We added in a production variable for prediction
- Both porosity and production have interesting relationships with production



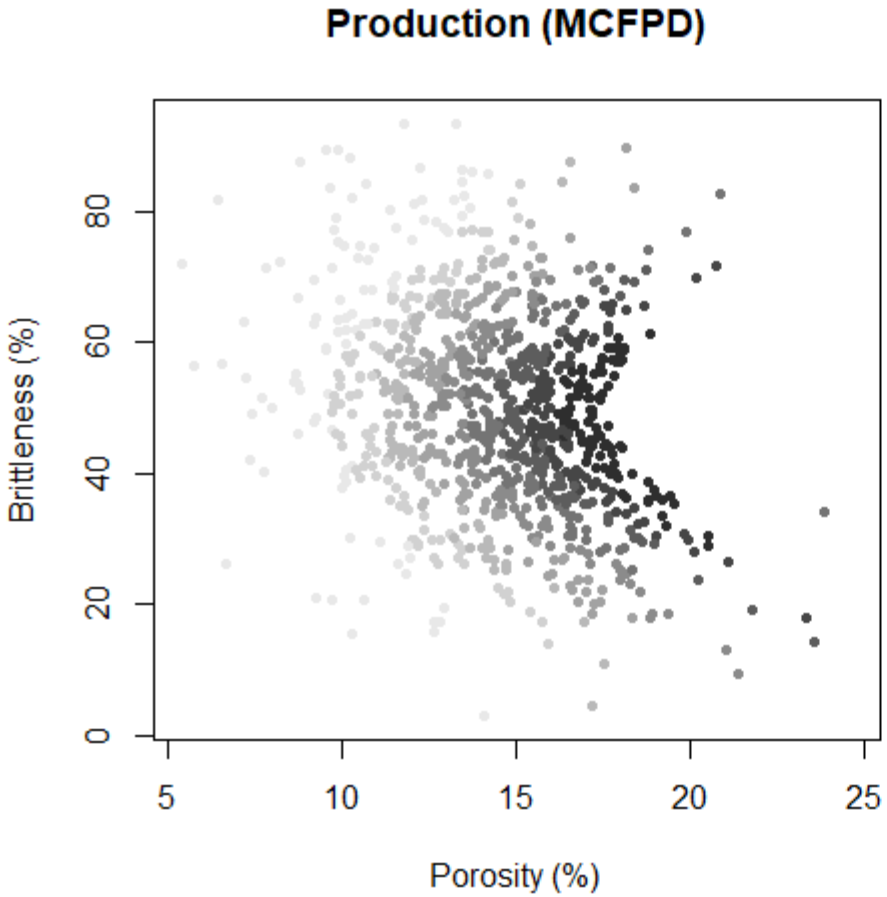


- To get a more complete story, check out the labeled scatter plot.

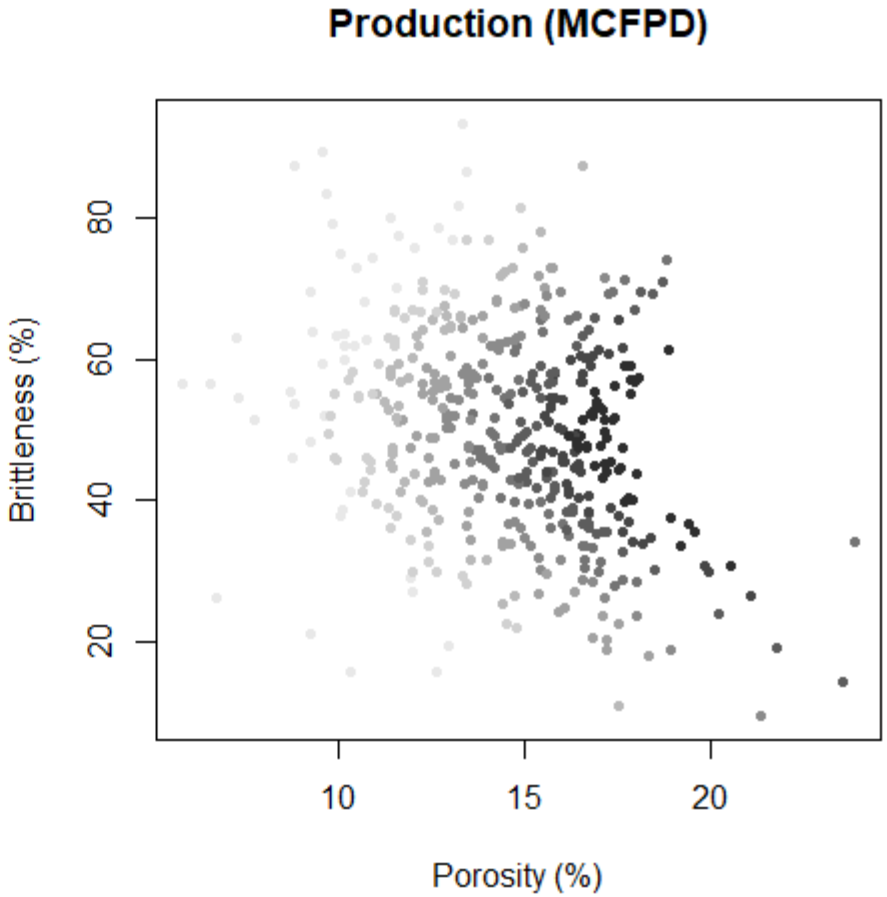
Decision Trees – Demonstration in R

Let's use our Unconventional Multivariate Data

- There is a complicated relationships between porosity, brittleness and production.



Available data set (n=500)



Training data set (n=500)

Decision Trees – Demonstration in R

Build the initial reasonably complicated tree

- By leaving the default tree controls on we get an 10 terminal node tree.
- We can use the summary command to:

```
Regression tree:
tree(formula = Prod ~ Por + Brittle, data = train, control = tree.control)
Number of terminal nodes:  10
Residual mean deviance:  302900 = 148400000 / 490
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2298.00  -303.50    57.16     0.00   327.50  3668.00
```

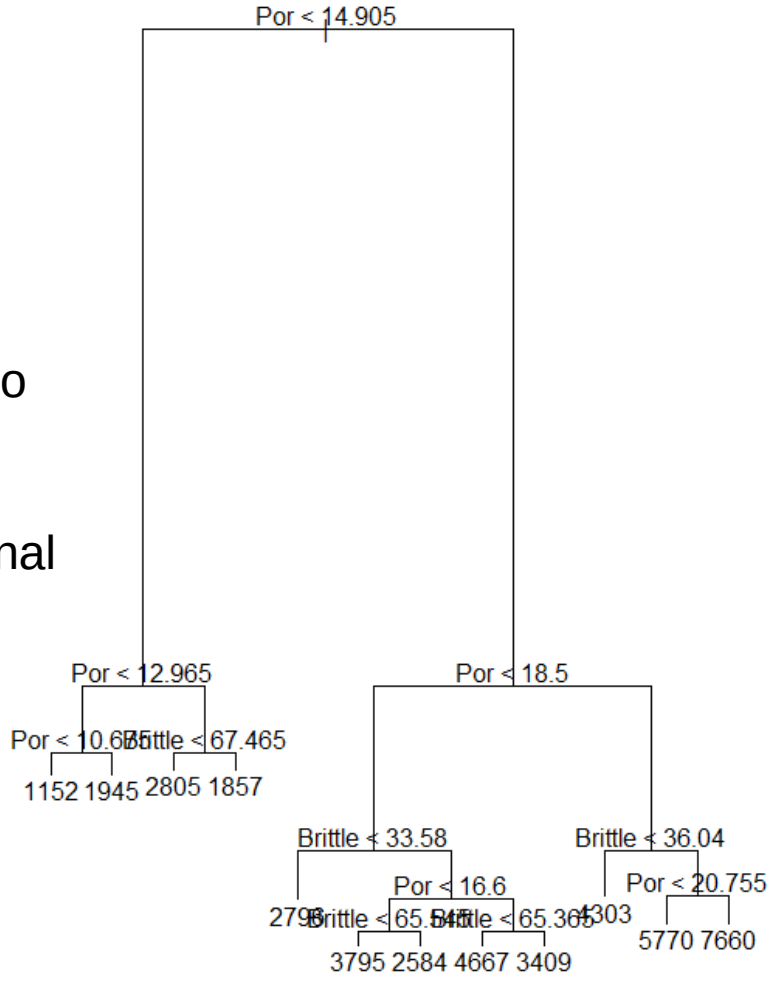
  - check the complexity of the resulting tree (number of terminal nodes)
  - check the summary statistics of the residuals and ensure that the model is not biased (mean = 0.0)
  - residual mean deviance is the total residual deviance divided by (the number of observations – number of terminal nodes)
  - for a regression trees the total residual deviance is the $\mathrm{RSS} = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$

Decision Trees – Demonstration in R

Build the initial reasonably complicated tree

Here's the tree:

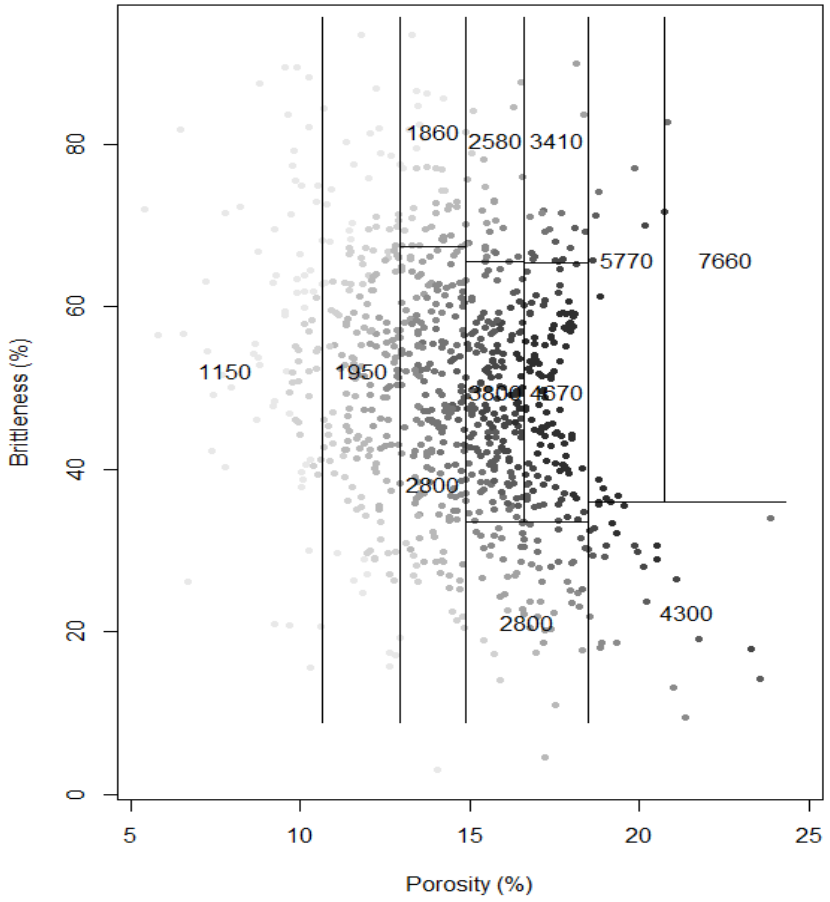
- first choice is porosity < or > 14.5%
- we get to the 3rd decision before brittleness
- is considered
- length of the branches is proportional to decrease in impurity
  - purity is a measure of the consistency in each region / terminal node.



Decision Trees – Demonstration in R

Build the initial reasonably complicated tree

We can plot the original data and the binary recursive boundaries outlining the various regions and the mean values in each region used as the estimate.

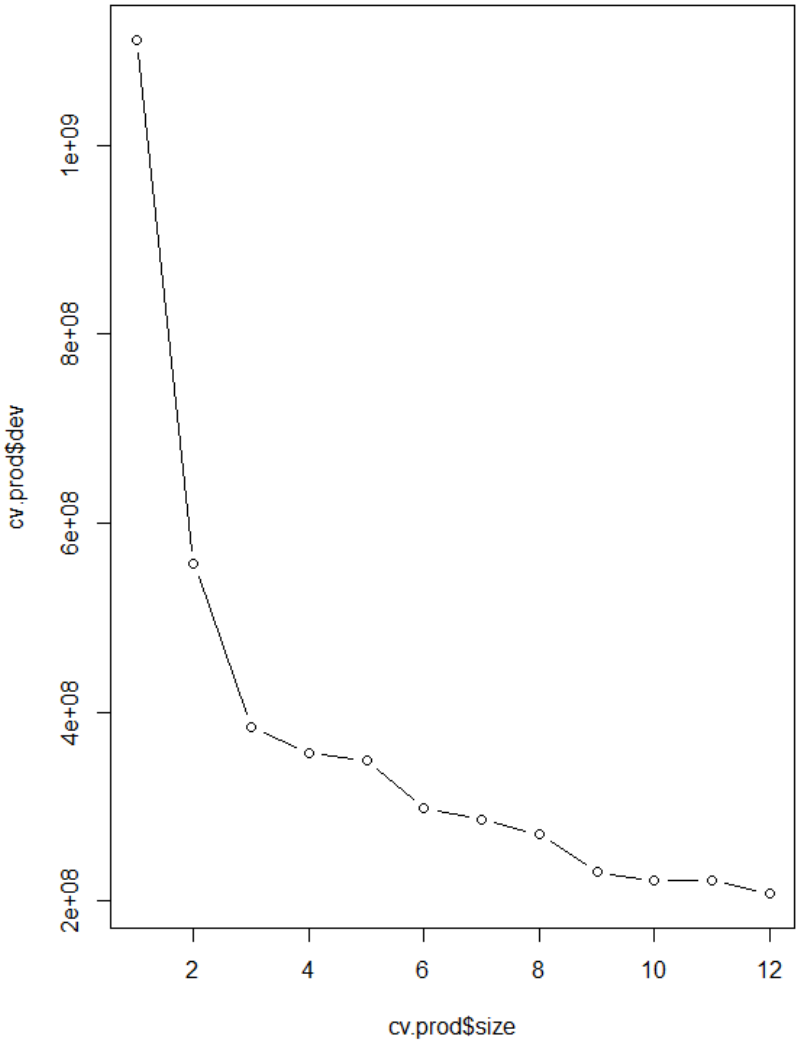


Decision Trees – Demonstration in R

Build the initial reasonably complicated tree

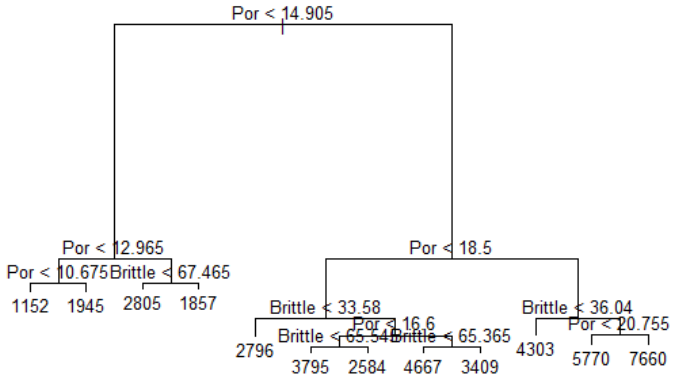
Then we perform k fold cross validation.

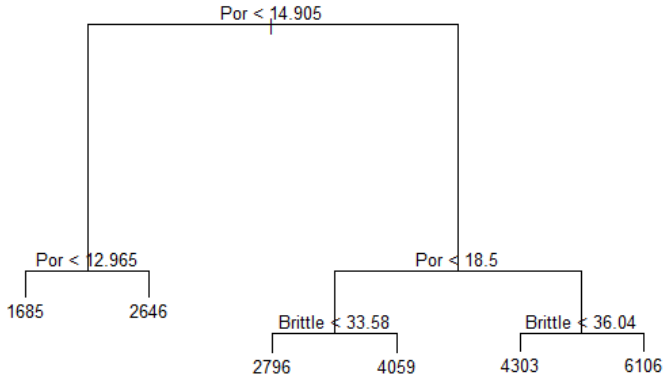
- Decrease tree complexity from 12 nodes (current model) to 1 node (uniform model)
- Calculate the RSS by averaging over k folds of the training data
- We can observed that each additional node improves the model
- We could simplify to 6 nodes and capture most of the benefit.

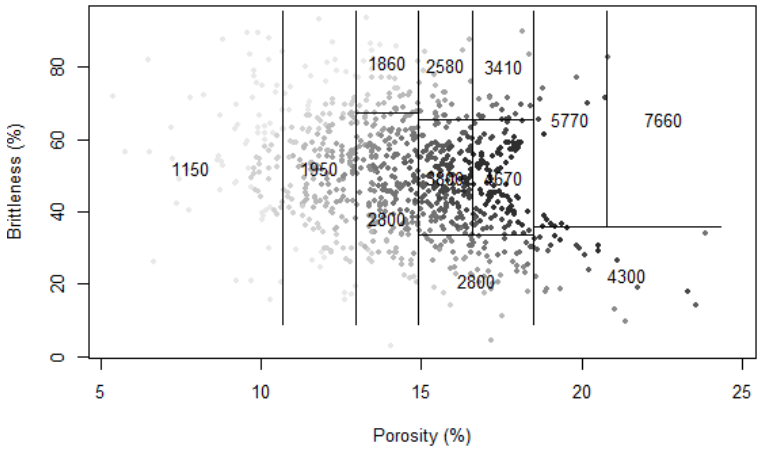


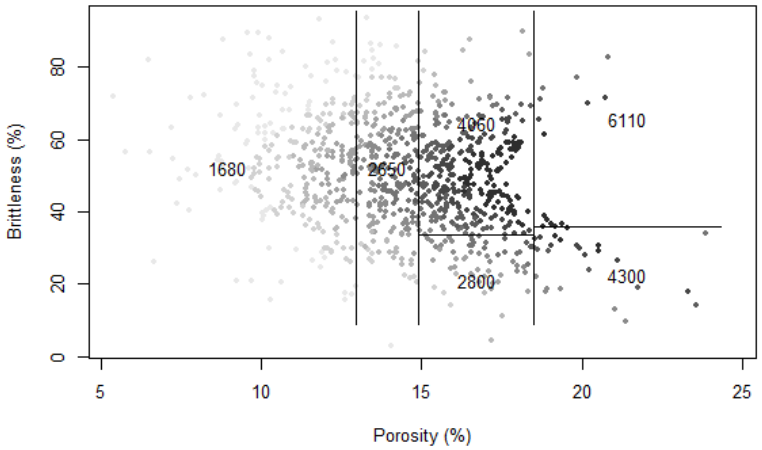
Decision Trees – Demonstration in R

Original and pruned tree:



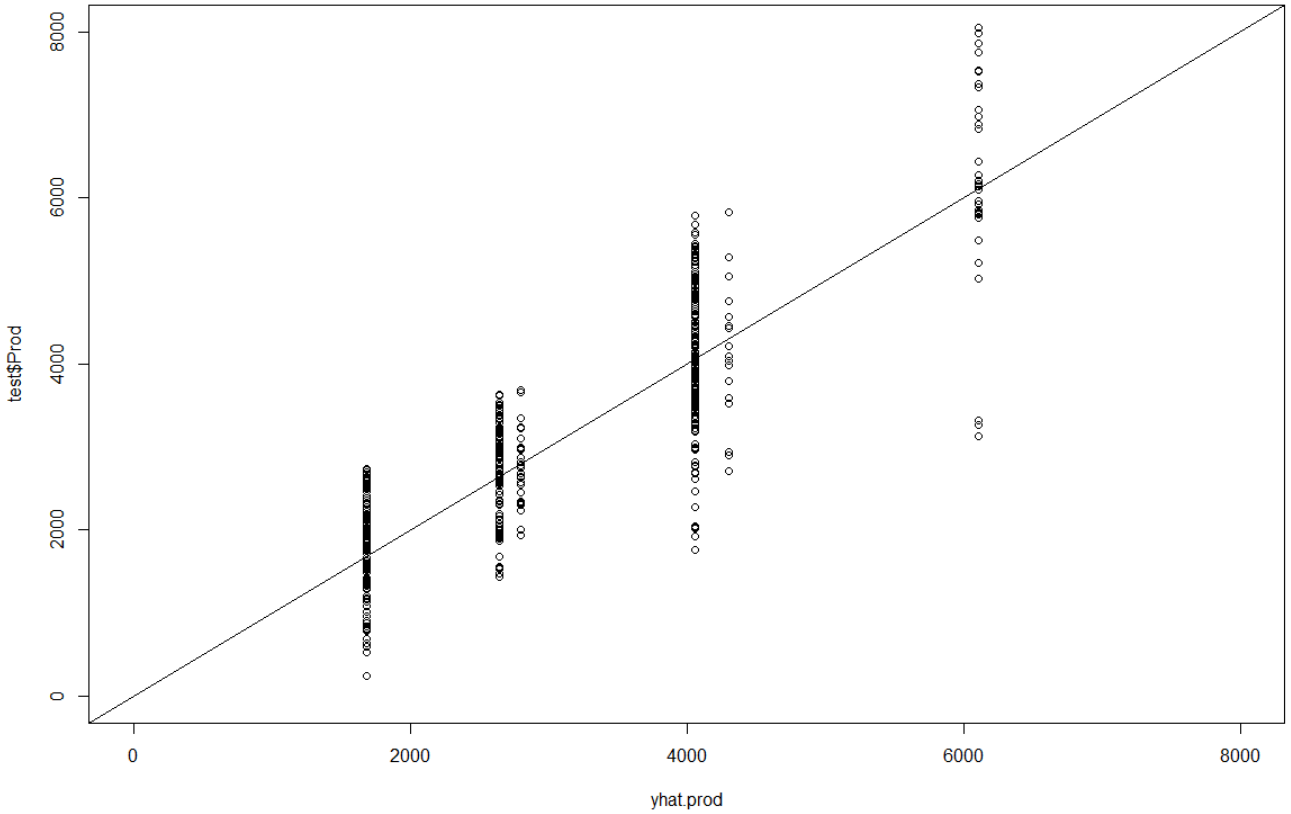






Decision Trees – Demonstration in R

Cross validation with the testing data set



- Note: the binning due to estimation with the mean of only 6 regions
- We can calculate MSE to assess model accuracy

Decision Trees – Hands-on in R

Build your own tree. Change the tree parameters to build more and less complicated trees and visualize solution space and prune the tree.

```
# We can design the controls on the tree growth
tree.control = tree.control(nobs = 500, mincut = 5, minsize = 10, mindev = 0.01)
# nobs is the number of data in training set, mincut / minsize are minimum node size constraints
# and mindev is the minimum deviation in a node to allow a split
# These of the defaults in the package. You can change these later and rerun to observe increased or
# decreased complexity.
```

- set mindev very small to increase complexity
- set mincut larger to decrease complexity

Decision Trees – Comments

General Comments on Decision Trees

- Easy to explain
- Analog to human decision making
- Graphically displayed
- Continuous or categorical variables

- Lower predictive accuracy than other machine learning methods
- Model variance may be high

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Bagging (use with many types of models)
  - the use of bootstrap on the training dataset to get B training sets
  - train a tree on each data set
  - then use all models and average the result to get the prediction

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

  - the trees are allowed to grow large
  - 100s to 1,000s of trees (forest of mediocre estimates!)
  - classification by majority vote
  - out-of-bag data (about 1/3 for each tree) are used as a test data set!

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Random Forest
  - same as bagging, but we randomize selection of on about $\sqrt{m}$ of the features!
  - prevents a single strong predictor form dominating the entire set of trees – forces diversity among the trees
  - decorrelating the trees

Bagging, Random Forest and Boosting

These are all methods to improve the prediction accuracy of trees

- Boosting (used with many types of models)
  - sequential modeling of a simple tree
  - build a tree, calculate residual
  - build a tree to model residual from 1st tree
  - build a tree to model the residual from 2nd tree
  - etc.

Machine Learning
New Tools

Topic	Application to Subsurface Modeling
Accuracy vs. Flexibility	Decide on model complexity *Build models that balance model bias and model variance to optimize complexity.*
Decision Trees	Use machine learning for modeling *Predict a difficult to measure / unavailable measure with a set of available measures. e.g. fill in missing data at wells or make a 2nd variable to assist with predicting porosity.*

Spatial Modeling with Geostatistics

Machine Learning

Lecture outline . . .

- **Estimation and Simulation**
- **Sequential Gaussian Simulation**

Prerequisites

Introduction

Probability Theory

Representative Sampling

Spatial Data Analysis

Spatial Estimation

Stochastic Simulation

Uncertainty Management

Machine Learning